

Claims

I claim:

1. A method for providing a status of a transaction with an application on a server, comprising:
 - receiving a request for a transaction on the server from a client;
 - invoking the application on the server to process the request;
 - generating a first polling code having a first Uniform Resource Locator (URL) that includes a first set of parameters, and sending the first polling code to the client; and
 - communicating the first URL from the client to the server to request the status of the transaction.
2. The method of claim 1, wherein the first set of parameters comprises a delay time and a polling count.
3. The method of claim 2, wherein the communicating step comprises communicating the first URL from the client to the server after expiration of the delay time.
4. The method of claim 2, wherein the communicating step further comprises communicating the first URL from a browser on the client to a server agent on the server after expiration of the delay time, and wherein the server agent determines the status of the transaction based on the URL.

5. The method of claim 2, further comprising:

generating, after the communicating step, a second polling code having a second URL that identifies a second set of parameters if the transaction is incomplete and the polling count does not equal a polling count limit;

sending the second polling code to the client; and

communicating the second polling code to the server from the client after expiration of the delay time to request the status of the transaction.

6. The method of claim 5, wherein the generating step comprises incrementing the polling count of the first set of parameters by one to yield a new polling count.

7. The method of claim 2, further comprising directing the client to an exception page after the communicating step if the polling count equals a polling count limit and the transaction is incomplete.

8. The method of claim 2, further comprising directing the client to a completion page after the communicating step if the transaction is complete.

9. A method for providing a status of a transaction with an application on a server, comprising:
- receiving a HTTP request for a transaction on the server from a browser on a client;
 - invoking the application on the server to process the HTTP request;
 - calling a client code generator to generate a first polling code having a first Uniform Resource Locator (URL) that includes a first set of parameters, wherein the first set of parameters comprises a delay time and a polling count;
 - sending the first polling code to the browser; and
 - communicating the first URL in the first polling code from the browser to a server agent on the server after expiration of the delay time to request the status of the transaction.
10. The method of claim 9, wherein the first URL is processed by the server agent on the server to determine the status of the transaction.
11. The method of claim 9, further comprising directing the browser to a completion page if the transaction is complete.
12. The method of claim 9, further comprising directing the browser to an exception page if the transaction is incomplete and the polling count equals a polling count limit.

13. The method of claim 9, further comprising generating a second polling code having a second Uniform Resource Locator (URL) that identifies a second set of parameters and the transaction requested if the transaction is incomplete and the polling count does not equal a polling count limit, wherein the second set of parameters comprises the delay time and a new polling count.

14. The method of claim 13, further comprising:

- sending the second polling code to the browser; and
- communicating the second URL in the second polling code from the browser to the server after expiration of the delay time to request the status of the transaction.

15. A system for providing a status of a transaction with an application on a server, comprising:

a server agent for receiving a request for the transaction from a client, and for initiating the transaction based on the request;

a client code generator invoked by the server agent for generating a first polling code having a first Uniform Resource Locator (URL) that includes a first set of parameters, and for sending the first polling code to the client, wherein the client communicates the first URL to the server agent to request the status of the transaction.

16. The system of claim 15, wherein the client comprises a browser for receiving the first polling code, and for communicating the first URL to the server agent.

17. The system of claim 15, wherein the first set of parameters includes a delay time and a polling count.

18. The system of claim 17, wherein the client communicates the first URL to the server agent after expiration of the delay time.

19. The system of claim 17, wherein the server agent processes the first URL to determine the status of the transaction, and invokes the client code generator to generate a second polling code having a second URL that identifies a second set of parameters if the transaction is incomplete and the polling count does not equal a polling count limit.

20. The system of claim 19, wherein the second set of parameters comprises the delay time and a new polling count, and wherein the client code generator increments the polling count of the first set of parameters by one to yield the new polling count.

21. A program product stored on a recordable medium for providing a status of a transaction with an application on a server, which when executed, comprises:

agent program code configured to receive a request for the transaction from a client, and to initiate the transaction based on the request;

generator program code invoked by the agent program code configured to generate a first polling code having a first Uniform Resource Locator (URL) that includes a first set of parameters, and to send the first polling code to the client, wherein the client communicates the first URL to the agent program code to request the status of the transaction.

22. The program product of claim 21, wherein the first set of parameters includes a delay time and a polling count.

23. The program product of claim 22, wherein the client comprises a browser configured to receive the first polling code, and to communicate the first URL to the agent program code after expiration of the delay time.

24. The program product of claim 22, wherein the agent program code is further configured to process the first URL to determine the status of the transaction, and to invoke the generator program code to generate a second polling code having a second URL that identifies a second set of parameters if the transaction is incomplete and the polling count does not equal a polling count limit.

25. The program product of claim 24, wherein the second set of parameters comprises the delay time and a new polling count, and wherein the generator program code increments the polling count of the first set of parameters by one to yield the new polling count.